UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/621,951 | 07/17/2003 | Mark Allen Grubbs | AUS920030463US1 | 1546 |

46073 7590 12/12/2007

IBM CORPORATION (VE)
C/O VOLEL EMILE
P. O. BOX 162485
AUSTIN, TX 78716

| EXAMINER | |
|---|---|
| LU, CHARLES EDWARD | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2161 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 12/12/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**



Application Number: 10/621,951
Filing Date: July 17, 2003
Appellant(s): GRUBBS ET AL.

Volel Emile
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 9/13/2007 appealing from the Office action mailed 4/13/2007.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

Duchamp, Dan. "Optimistic Lookup of Whole NFS Paths in a Single Operation." USENIX Summer 1994 Technical Conference, June 6-10, 1994. 13 pages.

Kleiman, Steve R. "Vnodes, an Architecture for Multiple File System Types in Sun UNIX." USENIX Summer 1986 Conference, June 1986. pp. 238-247.

Stern, Hal. "A File by Any Other Name." SysAdmin, SunWorld Online, September 1995. 9 pages.

| | | |
|---|---|---|
| 5,437,029 | SINHA | 7-1995 |
| 5,499,358 | NEVAREZ | 3-1996 |

Applicant Admitted Prior Art in application 10/621,951.

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

**9a. Claims 1, 4-5, 7-8, 11-12, 14-15, and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duchamp ("Optimistic Lookup of Whole NFS Paths in a Single Operation") evidenced by Kleiman ("Vnodes: An Architecture for Multiple File System Types"), in view of Sinha (U.S. Patent 5,437,029) of record, further in view of Applicant Admitted Prior Art (AAPA).**

**As to claim 1,** Duchamp teaches the following claimed subject matter:

Determining at least one file system object in a file system upon mounting the file system at a mount point (2.2.2) on a computer system (2.1, 2.2), each file system object having a pathname and an inode (Abstract, 2.1), the inode for locating the file system object on a storage system (sec. 1, 2); Note from Kleiman that a vnode is a file system-independent inode (Kleiman, p. 2). Kleiman was not cited as prior art, but only as evidence that a "vnode" is well-known in the art as reading on the inode as claimed.[1]

Entering the path name of the at least one file system object into a memory system (2.1);

Cross-referencing the path name of the at least one file system object in the memory system with its inode thereby enabling the inode to be obtained with one memory access (Abstract, 2.1).

Duchamp does not expressly teach "inode number".

However, Applicants admit, "an inode is identified by a unique number called an inode number" (p. 1, l. 25).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Duchamp with the above, such that in a lookup cache above the path is cross-referenced to the inode number. The motivation would have been to successfully access a file by using an identification number of the inode.

Duchamp as applied above does not expressly teach a "frequently accessed object."

However, Sinha teaches allowing a user to specify a frequently accessed object (col. 7, l. 26).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to further modify Duchamp with the above, such that frequently accessed objects specified by the user are additionally determined. The motivation would have been to allow the user to customize file system performance as taught by Sinha (col. 8, ll. 24-30).

**As to claim 4,** Sinha as applied above further teaches wherein determining includes the step of obtaining the path name from a user (see above). Furthermore, it is noted that Duchamp must obtain a path name from a user to store the path name (e.g., 2.1).

**As to claim 5**, Duchamp as applied above further teaches wherein determining includes monitoring accesses to a file system object within a certain time span (2.2.4)

**As to claim 7,** Duchamp as applied above further teaches wherein a path name of a file system object and its cross-referenced inode number is removed from the memory system when a user so ordered (see e.g., 2.2.4). Furthermore, it is noted that Sinha as applied above also teaches or suggests the claimed subject matter (e.g., col. 8, ll. 8-15).

**Claims 8, 11-12, 14-15, and 18-19** are drawn to substantially the same subject matter as claims 1, 4-5 and 7, discussed above.

---

[1] It should be noted that Kleiman is used only as evidence that the term "vnode" reads on the claimed "inode."

**9b. Claims 6, 13, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duchamp ("Optimistic Lookup of Whole NFS Paths in a Single Operation") evidenced by Kleiman ("Vnodes: An Architecture for Multiple File System Types"), in view of Sinha (U.S. Patent 5,437,029) of record, further in view of Applicant Admitted Prior Art and further in view of Stern ("SysAdmin : A file by any other name").**

**As to claim 6**, Duchamp, Sinha and AAPA as applied above do not expressly teach wherein the path name and its cross referenced inode number are removed from the memory system when the file system containing the file system object is unmounted.

However, Stern teaches that unmounting a file system purges the entries of a lookup cache (p. 4). Duchamp and Sinha are both drawn to lookup caches, as seen above, and as modified above, the lookup cache contain frequently used path names and their inode numbers.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Duchamp, Sinha, and AAPA with the above, such that the cache is purged when the file system is unmounted (and perform the claimed subject matter). The motivation would have been to clean up the memory system, since the cache entries would be invalid when the file system is unmounted.

**Claims 13 and 20** are drawn to substantially the same subject matter as claim 6, discussed above.

**9c. Claims 2-3, 9-10, and 16-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duchamp ("Optimistic Lookup of Whole NFS Paths in a Single Operation") evidenced by Kleiman ("Vnodes: An Architecture for Multiple File System Types"), in view of Sinha (U.S. Patent 5,437,029) of record and further in view of Applicant Admitted Prior Art and further in view of Nevarez (U.S. Patent 5,499,358).**

**As to claim 3,** Duchamp, Sinha and AAPA as applied above teaches a mounted file system as discussed above, but do not expressly teach an extended attribute file being associated with the mounted file system.

However, Duchamp teaches an extended attribute file in a file system (figs. 1-3).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to further modify Duchamp, Sinha and AAPA with the above, such that an extended attribute file is associated with the mounted file system. The motivation would have been to allow file system administrators to easily manage files and directories in a computer environment through the extended attributes, as taught by Nevarez (col. 5, ll. 6-11).

Duchamp, Sinha, AAPA, and Duchamp as applied above do not expressly teach obtaining from an extended attribute file a list of pathnames to be entered into the memory system.

However, in Sinha, there could be several file system objects (e.g., files) that the user desires to enter into the memory system (col. 7, ll. 20-29). As discussed by Sinha, file system objects comprise pathnames (e.g., Sinha, col. 7, ll. 40-50).

Furthermore, Nevarez teaches storing a database of user-defined data in the extended attributes of a file system (Abstract, Summary), and using this database to manage file system objects that have been tagged by a user (col. 4, l. 50 – col. 5, l. 32). For example, directory names matching a tag can be obtained (col. 5, ll. 1-34).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Duchamp, Sinha, and AAPA with the above, such that a list of pathnames to be entered into the system is obtained from the extended attribute database discussed above, the database storing user-tagged information to facilitate management of files and directories as discussed above. The motivation would have been to allow file system administrators to easily manage files and directories in a computer environment through the extended attributes, as taught by Nevarez (col. 5, ll. 6-11), for the files to be entered into the system in Sinha (col. 7, ll. 20-50).

**As to claim 2**, the combination as applied above would further teach or suggest wherein the pathnames in the extended attribute file are relative to the mount point, at least because the file system was discussed above as being mounted on a mount point, and paths are shown in Duchamp as relative to the mount point (see above, Duchamp, 2.2.2).

**Claims 9-10, and 16-17** are drawn to substantially the same subject matter as claims 2-3, discussed above.

**(10) Response to Argument**

**Claims 1, 8, and 15**

Appellant argues that the term "vnode" does not read on the claimed "inode" because the "vnode" is the independent part of the inode rather than both the independent and dependent part of the inode (Appeal Brief, p. 6, para. 2). The examiner respectfully disagrees.

The claims do not require that an inode comprises both independent and dependent parts. Furthermore, no such definition of "inode" is found in the specification. Therefore, the broadest reasonable interpretation was applied for the claimed "inode."

Duchamp teaches, "a single path lookup suffices to translate a path to a vnode" (sec. 2.1). Thus, Duchamp teaches a "vnode." Furthermore, Kleiman teaches, "The file system independent inode was renamed vnode." Thus, the "vnode" in Duchamp is a "file system independent inode." It should be noted that Kleiman was used only as evidence that a "vnode" of Duchamp is a "file system independent inode."

Therefore, the "vnode" of Duchamp reads on the claimed "inode" for the above reasons.

Appellant further argues that the combination of Duchamp, Kleiman, Sinha, and AAPA would not teach the step of 1) determining at least one frequently accessed file system object in a file system upon mounting the file system at a mount point on a computer system, 2) each file system object having a pathname and an inode number, 3) the inode number for locating the file system object on a storage system (Brief, p. 7, last paragraph), and there is "no reason for one skilled in the art to combine the

teachings . . . absent a specific teaching or suggestion to do so in the references (Brief, p. 7, first para.). The examiner respectfully disagrees.

In response to Appellant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the motivation was found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art.

It should be noted that Duchamp, Kleiman, and AAPA teaches or suggests most of the three limitations above, and Sinha was used to make up for the deficiency of a "frequency accessed" object. See prior Action, p. 3. For example, Duchamp as applied in the Prior Action teaches a substantial portion of limitation 1) above because Duchamp determines at least one file system object in a file system upon mounting the file system at a mount point on a computer system (2.1, 3rd paragraph and last paragraph, 2.2, 2.2.2, 2nd paragraph). Furthermore, Duchamp, Kleiman, and AAPA as applied in the Prior Action teaches or suggests all of limitations 2) and 3) above (see Prior Action). The motivation to combine is seen in the Prior Action on p. 3.

As discussed above, Duchamp, Kleiman, and AAPA teaches all of the claimed subject matter except a "frequently accessed" object. Thus, Sinha was applied to teach

a "frequently accessed" object (col. 7, l. 26). The combination was made so that Duchamp, Kleiman, and AAPA as applied above would support "frequently accessed" objects. The motivation would have been to allow a user to customize file system performance, as taught by Sinha (col. 8, ll. 24-30) and seen in the Prior Action, p. 3. As such, Duchamp, Kleiman, AAPA, and Sinha teach or suggest all of the claimed subject matter.

Appellants further state that "if one were to combine the teachings . . . the resulting combination would teach entering frequently-accessed files in the. . . cache of Duchamp. It would not, however, teach the step of determining at least one frequently-accessed file system object . . . ." The examiner respectfully disagrees. Entering frequently accessed files must include "determining at least one frequently accessed file system object" so that proper data can be entered. See above and Prior Action. Note that the term "file system object" is given its broadest reasonable interpretation of "a file, directory, etc., in a file system."

Therefore, the references would teach or suggest all of the claimed subject matter, and there would be a reason for one skilled in the art to combine the teachings of the references. Furthermore, the teaching or suggestion was found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art.

For the above reasons, the rejection of claims 1, 8, 15, and all other claims in this group should be sustained.

**Claims 2, 9, and 16**

Appellant argues that Nevarez does not teach, "Obtaining from an extended attribute file a list of pathnames to be entered into the memory system, the extended attribute file being associated with the mounted file system." The examiner respectfully disagrees.

Appellant appears to be arguing Nevarez individually. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In the Prior Action, the combination of all the references teaches all of the claimed subject matter (p. 6).

Appellant further argues that Nevarez cannot be combined with the other references to show the claimed invention, and the Examiner has impermissibly combined the references (Brief, p. 9, 4th paragraph). Appellant states that Sinha teaches that a user determines whether a file will be frequently accessed, and that the user supplies the path name (Brief, p. 9, 3rd paragraph). Appellants further state that if the user supplies the path name, there is no reason to obtain the name of the file from an extended attribute file, and thus, Nevarez cannot be combined with the other references (Brief, p. 9, 4th paragraph). The Examiner respectfully disagrees.

Nevarez was combined with the other references so that the user can obtain the pathname from Nevarez's extended attribute file and then supply the pathname to Sinha's system (which accepts pathnames). See Prior Action, p. 6. The motivation is

also seen in the Prior Action, p. 7. Thus, even if the user supplies the path name, there is reason to first obtain the name from a file before supplying it.

Therefore, there is a valid reason to combine Nevarez with the other references. Furthermore, Nevarez can be combined with the other references to teach or suggest all of the claimed subject matter, and the combination is permissible.

For the above reasons, the rejection of claims 2, 9, 16, and all other claims in this group should be sustained.

**Claims 6, 13, and 20**

Appellant argues that Stern does not teach the claimed subject matter (Brief, p. 10, paragraphs 2 and 6). The examiner respectfully disagrees.

Appellant appears to be arguing Stern individually. One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

The combination was made so that the cache will be cleared when the file system is unmounted (Prior Action, p. 5). As pointed out by Appellant, Stern teaches removing any cached entries for files on a file system that is unmounted (Brief, p. 10, para. 3). The other references already teach or suggest cached entries for path names and inode numbers (Prior Action, p. 5). Thus, the combination would clear the cached path name and inode number on a file system that is unmounted, and all of the claimed subject matter would be implemented. The motivation is seen in the Prior Action, p. 5.

For the above reasons, the rejection of claims 6, 13, 20, and all other claims in this group should be sustained.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Charles E. Lu
Assistant Examiner
AU2161
11/30/2007

Conferees:

Apu Mofiz
Supervisory Patent Examiner

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER

Eddie Lee
Supervisory Patent Examiner